AMENDMENT TO THE CLAIMS:

The following listing of claims will replace all prior versions of claims in the application:

Claims 1 – 35 (Canceled)

36. (New) A method of automatic generation, by means of a data processing system associated with a program called a Configurator for creating a global simulation model of an architecture comprising models of integrated circuits under development that can constitute, with the help of the automatic Configurator, a machine or a part of a machine, and environment simulation models that make it possible to test and verify the circuit under development, a configuration definition file for components of the architecture, these components constituting fixed functional blocks for describing the functionalities of integrated circuits or parts of integrated circuits, the components being chosen by the user from a library of various component types and a library of environment components, in order to create the global model of the architecture corresponding to the functional specification defined in the configuration definition file and conforming to the specification of the architecture of the global model specified by an architecture description file, comprising:

- reading an architecture description file of the global model and storing, in a component and connection rule table, in a connection coherency rule table, and in a source file formatting table, information related to all of the possible configurations, with each component being assigned a name that unambiguously identifies its position in the architecture, and a type from among several types (such as Active Components, Monitoring and Verification Blocks, Intermediate Blocks, System Blocks and Global Blocks);

- instantiating the components specified in the configuration definition file by a user-developer using a list of the present components , designated by their names and types and including parameters or invoking procedures, the configuration definition file comprising a file from which components are selected together with their types and optional additional indications concerning the type of interface and the server involved in the configuration to be generated by the Configurator, and storing the corresponding information in an instance connection table;

- topologically connecting the instances and storing the corresponding information in the instance connection table;

- physically connecting the interface signals, at the level of each instance of the components, by applying regular expressions, stored in the component and connection

rule table, based on the names of the signals constituting a wiring table; and

- using the instance connection table, the wiring table and the formatting table to automatically generate HDL-type and HLL-type source files of the global simulation model corresponding to a configuration specified by the configuration definition file

37. (New) A method according to claim 36, wherein the Configurator transmits to the HLL-type parts of each component information on:

- the name (LABEL) of the component;
- the type of the instance (DUT, XACTOR, VERIFIER, MONITOR); and
- an HDL path, comprising a hierarchical name of the component in the description of the model.

38. (New) A method according to claim 36, wherein the configuration definition file also includes a keyword identifying a server by name or number in which a component is instantiated when the method is used in a multi-server system.

39. (New) A method according to claim 37, wherein the configuration definition file also includes a keyword identifying a server by name or number in which a component is instantiated when the method is used in a multi-server system

40. (New) A method according to claim 39 wherein, in the case of a multi-server utilization, executed by the Configurator system comprising:

- dividing the Configuration into several HDL-type and HLL-type components;
- sorting the HDL-type components and the HLL-type components according to the servers to which they belong;
- generating the peripheral components of HDL-type used for sending and receiving signals between the parts of the configuration;
- duplicating the Global Blocks by the Configurator system and the instantiation of the Global Blocks duplicated in each server; and
- generating the HLL-type components that serve as a communication medium between the servers.

41. (New) A method according to claim 38, wherein the automatic connection between

the components by the Configurator is comprised of:

a wiring phase including the steps of:

– selecting the components and their respective positions; and

– creating the actual connection between the components,

said wiring phase generating as a result a wiring table that associates the signals connected to one another with the unique name of the wire that connects them; and

– generating source files for the HDL-type and HLL-type components.

42. (New) A method according to claim 41, wherein the wiring phase is performed by the Configurator system, and further comprises:

a. connecting the Global Blocks and the System Blocks to all of the components;

b. connecting the signals between the other components;

c. making an additional pass to connect the remaining unconnected signals of each component to predetermined signals in order to produce a given stable state; and

d. generating, via the Configurator system, partial configurations comprising a subset of the architecture.

43. (New) A method according to claim 42, wherein the predetermined signals are the signals of the System Block corresponding to the component.

44. (New) A method according to claim 36, wherein the architecture description file of the global model includes simulation models of Global Blocks and System Blocks, said Global Blocks and System Blocks being connected to one another and adapted for handling environment signals.

45. (New) A method according to claim 44, wherein the System Blocks are connected to other components and supply them with system signals that are specific to said other components.

46. (New) A method according to claim 45, wherein the data processing system performs a conformity check of the connections, comparing the connection table of the real instances between blocks to the connection coherency rule table.

47. (New) A method according to claim 46, wherein the data processing system compares the physical connections between the components to the connection coherency rule table, in order to detect any incompatibilities between the ends of the connections between the components, and in cases where an incompatibility is detected, the data processing system specifies and adds an adapter component (Intermediate Block) to the instance connection table, said adapter component is inserted into the incompatible connection between the components.

48. (New) A method according to claim 47, wherein the configuration definition file includes information, specified by an attribute, concerning the utilization of adapter components (Intermediate Blocks) with the instances of the active Components, whose connections are compared to the instance connection table, in order to detect any incompatibilities between the ends of the connections between the components, and in cases where an incompatibility is detected, the data processing system specifies and adds an adapter component (Intermediate Block) to the instance connection table, said adapter component is inserted into the incompatible connection between the components.

49. (New) A method according to claim 48, wherein the data processing system selects certain connections between the components of the connection coherency rule table, specifies additional connections constituting branches leading to respective additional models, which represent tools (probes) for monitoring the connections, and adds said additional connections to the instance connection table.

50. (New) A method according claim 36, wherein the Configurator system includes a source file generation phase, in which the HDL-type and HLL-type source files are generated based on the content of the component and connection rule table, the coherency rule table, the source file formatting table, the instance connection table and the wiring table.

51. (New) A method according to claim 50, wherein the data processing system executes an operation through the Configurator system for each configuration variant, in order to obtain several simulation models corresponding to the same functional specification, written in a description comprising various mixtures of languages of different levels (HDL, HLL).

52. (New) A method according to claim 36, wherein the data processing system generates a functional specification of the global simulation model in a computer format compatible with a high-level programming language, and in a format compatible with a hardware description language.

53. (New) A method according to claim 52, wherein the configuration definition file comprises, for each component, at least one part in HDL-type language, said part in HDL-type language providing an interface with other models.

54. (New) A method according to claim 53, wherein the models that include a part in HLL-type language include interface adapters.

55. (New) A method according to claim 54, wherein the Configurator system chooses each interface adapter model as a function of the connection coherency rule table.

56. (New) A method according to claim 55, wherein the connections of the physical signals are specified by "Ports," each port being an arbitrary selection of the signals of the HDL-type interface of a component by means of regular expressions based on the names of said physical signals, and being constituted by regular expression/substitute expression pairs, said expressions being successively applied to the name of each signal of the HDL-type interface, and if the final substitution is identical for two signals, the signals are connected to one another, the connection being stored in the wiring table.

57. (New) A method according to claim 56 wherein, each interface adapter being shared among several models connected to the same port, and only one of said models transmits signals through said port.

58. (New) A data processing system for automatically generating a global simulation model of a configuration of fixed functional blocks, mutually connected by inter working connections so as to constitute the global simulation model of an architecture comprising models of integrated circuits under development that can constitute a machine that conforms to the functional specification of a configuration, comprising a Configurator program having means for creating a simulation of wiring by applying stored regular expressions, and for using a configuration definition file in a high level language, a component and connection rule table describing properties of software components for simulating the circuit, a connection coherency rule table in a high level language, means for instantiating the components resulting from the configuration definition file, and an HLL code generator that combines the parameters of the components with the connection rules.

59. (New) A system according to claim 58, wherein there are at least five types of components: Active Components, Monitoring and Verification Blocks, Intermediate Blocks, System Blocks and Global Blocks.

60. (New) A system according to claim 59, further comprising means to perform a conformity check of the connections by comparing the instance connection table with a table of coherency rules for the physical connections between the models chosen from the blocks constituting the global model.

61. (New) A system according to claim 60, further comprising means to compare the instance connection table to the connection coherency rule table, in order to detect any incompatibilities between the ends of the connections between blocks, and in cases where an incompatibility is detected, the data processing system specifies and adds an adapter component (Intermediate Block) to the instance connection table, said adapter component is inserted into the incompatible connection between the components.

62. (New) A system according to claim 58, wherein the component and connection rule table, which includes the properties of the components, contains global parameters common to all of the component types and exists in the form of a table distributed into one or more tables, which may be associative, wherein the entries are names designating all of the possible models for the same component.

63. (New) A system according to claim 62, wherein the associative tables are adapted to contain the description, either in the form of parameter sets or in the form of references to procedures that generate the required values, the entries of these associative tables being names designating all of the possible models for the same component and forming a character string containing predetermined special identifiers, replaced by values calculated by the Configurator system.

64. (New) A system according to claim 63, wherein at least three selectors indicate the instance to be used, the following selectors are transmitted as parameters to a constructor of an HLL object:

- a first selector indicating the current instance ("item");
- a second selector specifying the instance connected to the other end of the port; and
- a third selector indicating the composite instance corresponding to the active Component containing the observation port.

65. (New) A system according to claim 58, wherein the Configurator system comprises one or more connection coherency rule tables, which represent the rules for interconnecting the components and for inserting intermediate components; one or more component and connection rule tables, which represent the system-level connection rules and the rules for generating connections between the signals; and one or more source file formatting tables, which represent the rules for generating instances of HLL-type objects.

66. (New) A system according to claim 58, wherein the Configurator system comprises:

- an HLL base class uniquely identifying each object instantiated and polling the configuration;
- means for generating and automatically instantiating System Blocks;

- means for using tables to associate the signals connected together under a unique name of the connecting wires; and

- means for using a formatting table to generate HDL-type and HLL-type source files.

67. (New) A system according to claim 58, wherein an operator functionally specifies the configuration in the highest level language to the extent possible, and completes the functional specification with the components in the lowest level language.

68. (New) A system according to claim 58, wherein the following entries in the hash define the Component Type (for example DUT (HDL model), XACTOR (transactor), MONITOR, VERIFIER or other types), and correlate each Component Type to a hash, in turn composed of the following entries:

– a first entry comprising the name of the HDL module of the component and the name of the corresponding source file; and

– a second entry comprising the definition of the method for selecting the signals that are part of a Port, this description being comprised of a set of entries indexed by the name of the Port; the configurator associates each Port name with a table of regular expressions and a pointer to a signal connection procedure that controls the application of these expressions to the names of the signals of the interface of the component.

69. (New) A system according to claim 68, wherein the Active Components have a generic structure that includes a Block containing the HDL description and a Block in HLL that provides the access paths to the HDL resources, and optionally, a description of the block in HLL; and the set of signals of the HDL block constitute the interface of the containing Block, formed by Ports, which are arbitrary logical selections of the signals of an interface, and by interface adapters, which are the software parts that handle, in each Port, the two-way communication between the parts in HLL and those in HDL, the interface adapters being chosen by the Configurator system.

70. (New) A system according to claim 69, wherein the Ports are specified in the form of regular expressions, which may serve both to select the subsets of signals to be connected and to define the connection rules.

71. (New) A system according to claim 58, wherein the Configurator system generates Transfer Components, which are inserted on each side of the interface between servers, these components simply being wires for the inputs and registers for the outputs.

72. (New) A method of automatic generation, by means of a data processing system associated with a program called a Configurator for creating a global simulation model of an architecture comprising models of integrated circuits under development, comprising:

- reading an architecture description file of the global model and storing information related to all of the possible configurations;
- instantiating the components and storing the corresponding information in an instance connection table;
- topologically connecting the interface signals,
- physically connecting the interface signals, at the level of each instance of the components using a component and connection rule table, and storing the corresponding information in a wiring table; and
- automatically generating the HDL-type and HLL-type source files of the global simulation model, corresponding to the configuration specified by the configuration definition file.

73. (New) A method according to claim 72, wherein the Configurator transmits to the HLL-type parts of each component information on:

- the name of the component;
- the type of the instance (DUT, XACTOR, VERIFIER, MONITOR); and
- an HDL path, comprising a hierarchical name of the component in the description of the model.

74. (New) A method according to claim 73, wherein the configuration definition file also includes a keyword indicating the name or number of the server in which a component is instantiated when the method is used in a multi-server system.

75. (New) A method according to claim 74, wherein the configuration definition file also includes a keyword indicating the name or number of the server in which a component is instantiated when the method is used in a multi-server system.

76. (New) A method according to claim 75 wherein, in the case of a multi-server utilization, executed by the Configurator system comprising:

- dividing the Configuration into several (HDL type and HLL type) parts;
- sorting the HDL-type components and the HLL objects according to the servers to which they belong;
- generating the HDL-type peripheral components used for sending and receiving signals between the parts of the configuration;
- duplicating the Global Blocks by the Configurator system and the instantiation of the Global Blocks duplicated in each server;, and

generating the HLL-type parts that serve as a communication medium between the servers.

77. (New) A method according to claim 72, wherein the architecture description file of the global model includes simulation models of Global Blocks and System Blocks, said Global Blocks and System Blocks being connected to one another and adapted for handling environment signals.

78. (New) A method according to claim 77, wherein the System Blocks are connected to other components and supply them with system signals that are specific to said other components.

79. (New) A method according to claim 78, wherein the data processing system performs a conformity check of the connections, comparing the connection table of the real instances between blocks to the connection coherency rule table.

80. (New) A method according claim 79 wherein, the Configurator system generates the source files in HDL language and in HLL language, in a source file generation phase, based on the content of the component and connection rule table, the coherency rule table, the source file formatting table, the instance connection table and the wiring table.

81. (New) A method according to claim 80, wherein the data processing system executes an operation through the Configurator system for each configuration variant, in order to obtain several simulation models corresponding to the same functional specification, but written in a description comprising various mixtures of languages of different levels (HDL, HLL).

82. (New) A method according to claim 79, wherein the data processing system generates a functional specification of the global simulation model in a computer format compatible with a high-level programming language, and in a format compatible with a hardware description language.

AMENDMENT TO THE ABSTRACT

Please substitute the following Abstract: